REMARKS

The above amendment and these remarks are responsive to the Office Communication of Examiner Victor D. Lesniewski, mailed 11/09/2004, and designated as non-Final.

Claims 1-53 are in the case, none as yet allowed.

35 U.S.C. 101

5. Claims 44-48 have been rejected under 35 U.S.C. 101 as directed to non-statutory subject matter.

Applicants have amended the claims as suggested by the Examiner to recite a computer readable medium.

35 U.S.C. 102

8. Claims 1-14, 16-27, 29-32, 34, and 36-53 have been

rejected under 35 U.S.C. 102(b) over Fiveash et al. (U.S. Patent 6,076,168, hereinafter Fiveash).

Applicants traverse, and argue that the Examiner has not established a prima facie case of anticipation.

Applicants invention relates to IP packet filtering which occurs in an operating system kernel implementation of, for example, the TCP/IP protocol suite. Access rules are expressed as filters referencing system kernel data; for outbound processing, source application indicia is determined; for inbound packet processing, a look-ahead function is executed to determine target application indicia; and responsive to the source or target application indicia, filter processing is executed

Claim 1 recites:

> "A method for control and management of communication traffic, comprising the steps of; expressing access rules as <u>filters referencing system kernel data</u>..." [Emphasis added].

With respect to claim 1, clause 1, Fiveash col 3 lines

21-25 are cited by the Examiner as anticipating. These lines, the referenced fig 4, plus lines 27-32 described the filters used by Fiveash. All the data cited is data that is contained in the IP packets that are sent and received. This is the traditional view of filters, and is conventionally represented in Fiveash. Conventional that is, prior to the filing of present application, which filters IP packets based on data not contained in the IP packets. The present application expands the conceptual basis for IP filtering, by extending such filtering to data not heretofore considered 'available' for filtering, that is, to system kernel data. Hence, Fiveash does not anticipate claim 1 of the present application.

Continuing, claim 1, clause 2 further recites:

> ".... for outbound processing, determining source application indicia...".

Further with respect to claim 1, clause 2, Fiveash col 3 lines 8-20 are cited by the Examiner as anticipating. Both Fiveash and the present application perform outbound filtering. The 'source application indicia' referred to in claim 1 clause 2 is the general terminology for non-IP packet data. The 'source application' is a reference to the non-kernel software that generated and sent the non-protocol data in the IP packet, and the term 'indicia' is a reference to any data associated with the application and not contained in the IP packet. Hence this clause concerns the identification and access of non-IP packet data, during or on behest of outbound processing.

However, Fiveash does not filter outbound packets based on source application indicia. Fiveash uses traditional IP packet filtering criteria which is data found in the IP packet (see Fiveash col 3, lines 21-32). This is also illustrated in Fig 4, and in col 4 lines 44-57. Hence, 'determining source application indicia' is not done by Fiveash, and is not anticipated by Fiveash.

Continuing claim 1, clause 3, further recites:

> "...for inbound packet processing, executing a look-ahead function to determine target application indicia...".

With respect to claim 1, clause 3, Fiveash col 2 line 61 through col 3 line 7 is cited by the Examiner. The citation describes broadly the processing of an inbound IP packet ("data packet"). Fig 2 is referenced. For example, "... it is determined whether the data packet is encrypted and/or whether authentication is required." These determinations are made based entirely on the data packet contents. This is apparent in Fiveash and because the technology context of Fiveash is the IP Security protocols AH and ESP. [These are protocols defined by the standards setting organization IETF. A partial list of references is the following RFC's, all easily available from the IETF web site; RFC2401, RFC2402 (AH), RFC2406 (ESP), RFC2408, RFC2409]. Once processing is done, the "packet is passed to the application layer." At no time does Fiveash execute a look-ahead function, and at no time does Fiveash determine target application indicia. Hence Fiveash does not anticipate this clause.

Continuing claim 1, clause 4, further recites:

"...and responsive to said source or target application indicia, executing filter processing."

With respect to claim1, clause 4, Fiveash col 2 lines 13-14 are cited by the Examiner. Filter rules can have a variety of actions. One possibility is to uses filters "to permit or deny acceptance of transmitted data", as in Fiveash. The present application does not focus on filter rule actions, but rather filter rule criteria. As we have seen, Fiveash does not determine, or consider, or mention 'source or target application indicia', hence whatever actions are taken, they cannot be taken 'responsive to said source or target application indicia.' Hence Fiveash does not have bearing on this clause. (The actions taken by the present application could be to permit or deny acceptance of transmitted data, or could be any other action.)

Claims 2-9 depend from base claim 1, and are all similarly distinguished from the Fiveash reference.

Further, claim 2 recites:

> "... further comprising the steps of executing said determining and executing steps within a kernel filtering function upon encountering a filter selector field referencing kernel data not included in said packet".

With respect to claim 2, Fiveash col 4 line 64 through col 5 line 22 is cited by the Examiner. The cited section describes the parameters allowed on two of the commands (genfilt & chfilt) in Fiveash. These parameters concern the data that is allowed in the filters used by Fiveash. As can be seen in col 5, lines 1-22, none of the filtering data is 'not included in said packet'. Note that a number of the parameters are used to control ancillary functions related to filtering; for example, logging control (-l), and the selection of which interface the filter is meant for (-i). These parameters are not part of the filtering process itself. Hence, Fiveash does not reference kernel data not included in the IP packet, and therefore Fiveash does not anticipate this claim.

Further, claim 3 recites:

> "...said filter processing including the steps of; determining a task or thread identifier; based on said task or thread identifier, determining a process or job identifier..."

With respect to claim 3, Fiveash col 1 lines 20-26 are cited by the Examiner. The cited text describes the Fiveash patent view of filtering in nonspecific terms, and actually takes a fairly narrow view, saying "Filtering is a function in which incoming and outgoing packets of data are accepted or denied...". Yes, but it can be much more. As mentioned above, the filtering action is quite distinct from the filtering process, and many actions are possible beyond just accepting or denying packets. Consider, for example, email spam filtering. Other aspects of filtering are much broader than Fiveash suggests, and one such aspect is contained in the present application. Fiveash's nonspecific description certainly does not include the steps in applicant's claim 3. Nowhere in the Fiveash patent are such steps mentioned, or even suggested. Hence Fiveash does not anticipate claim 3.

Continuing, claim 3 further recites:

> "... based on said process or job identifier, determining job or process attributes for filter processing".

Fiveash col 5, lines 1-22 are cited by the Examiner. This citation has been considered above; it describes the parameters on commands used to create Fiveash filters. It is quite apparent these parameters do not include a 'process or job identifier', hence these cannot be part of the filter processing used by Fiveash. Hence Fiveash does not anticipate the use of process or job identifier for filter processing.

Continuing with respect to the other claims depending from base claim 1, which has been distinguished from Fiveash above, additional claim limitations of claims 4-7 not taught by Fiveash include the following:

Claim 4: "... determining a user identifier; and based on said user identifier, determining user attributes...".

This is specific instance of a kind of non-IP packet data.

Claim 5: "... determining from said task identifier a work control block containing said process or job identifier..."

This is a specific instance of a kind of non-IP packet data.

Claim 6: "...passing an inbound packet to a sockets layer to identify said target application".

This is part of the look-ahead function.

Claim 7: "... marking said inbound packet...".

This is an aspect of the look-ahead function.

Claim 10 recites:

"A method for control and management of aspects of communication traffic within filtering, comprising the steps of: receiving IP packet data into a TCP/IP protocol stack executing within a system kernel ... executing filtering code within said system kernel with respect to non-IP packet data accessed within said system kernel outside of TCP/IP protocol stack ...".

With respect to claim 10, Fiveash Fig 1, items 110 & 120 are cited by the Examiner. Yes, applicant's invention uses the IP stack and assumes something like a 'filter module. However, the cited figure items make no mention of 'non-IP packet data accessed with said system kernel'. No where in the Fiveash patent is such an idea mentioned. Hence Fiveash does not anticipate this claim.

With respect to claims 11-14, and 16-17, these claims all depend from claim 10, and are distinguished from Fiveash as discussed above. Further limitations in these claims distinguishing Fiveash are as follows:

claim 11: "... said non-IP packet data including ..."

Claim 12: "... said non-IP packet data including..."

Claim 13: "... said non-IP packet data including..."

Claim 14: "... said context data..."

Claim 16: "... accessing filtering attributes..."

This refers to IP and non-IP packet data.

Claim 17: "... accessing filtering attributes..."

Claim 18 recites:

> "A method for centralizing system-wide communication management and control within filter rules, comprising the steps of: providing filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values; and said selector referencing data that does not exist in the IP packets."

With respect to claim 18, the Examiner cites Fiveash Fig 4 and col 3 lines 21-32. Fig. 4 is a list of example filter rules (in a textual form), showing specific IP packet data that is the subject of actual filtering (e.g. IP addresses, port numbers, transport protocol, etc.). The cited text explains the rules and fields in more detail. Typical is "Rules 12 through 17 filter outbound FTP ... service from addresses 10.0.0.1 and 10.0.0.5 through tunnel 4." (Col 3, lines 51-53). Missing entirely from the cited portions of Fiveash, and in Fiveash overall, is anything like "referencing data that does not exist in the IP packets." Hence, Fiveash does not anticipate this claim.

Claims 19-21 depend from claim 18, and are similarly distinguished.

Further with respect to claim 19, the claim recites:

> "...selectively including userid, user profile, user class, user group, user group authority, user special authority, job name, process name, job group, job class, job priority, other job or process attributes..."

These are all instances of non-IP packet data, and as previously explained, such are not taught by Fiveash for filtering.

Further with respect to claim 21, the claim recites:

> "... selectively to job indicia", and

> "... referencing kernel data not included in said traffic."

Neither of these limitations are taught by Fiveash.

With respect to claim 22, unlike earlier claims, this claim does refer to an action, namely, 'disallow selective IP packet traffic'. However the steps leading to this action are not contained in Fiveash. In particular, the following are not in Fiveash, in any of the cited sections or anywhere else;

'executing a look-ahead function' (clause 1),

'to determine a target application' (clause 1),

'processing said packet by determining a task ID' (clause 4),

'responsive to said task ID, determining a corresponding work control block' (clause 5),

'determining a user ID, process or job identifier from said work control block' (clause 6),

'from the user ID, process or job identifier selectively determining attributes for said user process or job' (clause 7), and finally, 'passing said attributes to said filter processor for managing and controlling communication traffic' (clause 8).

Hence Fiveash does not anticipate this claim.

Claims 23 and 24 are distinguished from Fiveish as containing recitations previously discussed. Specific claim limitations of claims 23 and 24 over Fiveish include:

Claim 23: "... said selector referencing data that does not exist in IP packets..."

Claim 24: "... determining a task ID; responsive to said task ID, determining a corresponding work control block; responsive to said work control block, determining a process or job identifier; responsive to said process job identifier, determining job or process attributes."

Claims 25-27 depend from base claim 24, and are distinguished from Fiveish as discussed above. Further specific claim limitations distinguishing these claims from Fiveish are as follows:

Claim 25: "... determine a target application for said packet..."

Claim 26: "... executing a look-ahead function..."

Claim 27: "... function to request of a sockets layer the identity of an application to which said sockets layer would pass said packet..."

Claim 29 recites:

"said kernel including a filter processor;

... determining a task ID; responsive to said task ID, determining a corresponding work control block;

determining a user ID control block determining attributes for said user;

and passing said attributes to said filter processor..."

*****

With respect to claim 29, the Examiner cites col. 3, lines 21-25 , col. 1, lines 20-26, col. 5, lines 12-22, and column 4, lines 36-39.

With respect to filtering based on system kernel data, Fiveash Col. 3, lines 21-25 are cited by the Examiner. These lines, the referenced Fig 4, plus lines 27-32 described the filters used by Fiveash. All the data cited is data that is contained in the IP packets that are sent and received. This is the traditional view of filters, and is conventionally represented in Fiveash. Conventional that is, prior to the filing of present application, which filters IP packets based on data not contained in the IP packets. The present application expands the conceptual basis for IP filtering, by extending such filtering to data not heretofore considered 'available' for filtering, that is, to <u>system kernel data</u>. Hence, Fiveash does not anticipate claim 29 of the present application.

With respect to determining attributes for filtering from the user ID control block, Fiveash col. 5, lines 1-22 are cited by the Examiner. This citation has been considered above; it describes the parameters on commands used to create Fiveash filters. It is quite apparent these parameters do not include a 'work control block' built from a task ID, hence these cannot be part of the filter processing used by Fiveash. Hence Fiveash does not anticipate the use of process or job identifier for filter processing.

Claims 30-32 depend from claim 29, and are similarly distinguished from Fiveash. Further distinctions include:

Claim 30: "... determine a target application..."

Claim 31: "... executing a look-ahead function..."

Claim 32: "... including the steps of operating a filter function to request of a sockets layer the identity of an application to which said sockets layer would pass said packet"

How these limitations distinguish the Fiveash reference has been discussed above.

With respect to independent claim 34, clauses 1 and 2, Fiveash col 2 line 61 through col 3 line 7 is cited by the Examiner. This citation has been discussed earlier. The citation concerns Fig 2 of Fiveash, which is a high-level (general) depiction of inbound IP packet processing for IP Security (see IETF references given above) that uses filters. Not mentioned in Fiveash anywhere is a filter "referencing non-packet data". For claim 34 clause 3 the same Fiveash citation is given. Not mentioned in Fiveash anywhere is "... responsive to said filter, executing a look-ahead function..." for any reason. In addition, Fiveash does not execute a look-ahead function "... for identifying a target application for said inbound packet".

With respect to independent claim 36, both the present application and Fiveash involve filtering of IP packets. And in both, this filtering activity occurs in an operating system kernel. But this claim has more, and despite the usual Fiveash citations, as previously discussed, the following limitations of claim 36 are not covered, or mentioned, or involved in any part of the Fiveash patent:

"... receipt of an outbound packet for determining a source application;" (clause 3),

"... receipt of an inbound packets processing for executing a look-ahead function..." (clause 4),

"...responsive to said source or target application for executing filter processing." (clause 5).

Claims 37-53 each contain limitations not taught by Fiveash, substantially as described previously with respect to one or more of claims 1, 10, 18, 22-245, 29, and 34. At least one distinguishing feature of each of these claims is set forth hereafter, as follows:

Claim 37: "... operable with respect to non-IP packets data..."

Claim 38: "...referencing data that does not exist in IP packets..."

Claim 39: "... executing a look-ahead function...", and more.

Claim 40: "... referencing data that does not exist in IP packets..."

Claim 41: "... determining a task ID..." and many more.

Claim 42: "...determining a task ID..." and many more.

Claim 43: "... a look-ahead function responsive to said filter..."

Claims 44 and 49: "... executing a look-ahead function..." and more.

Claims 45 and 50: "... with respect to non-IP packet data ..."

Claims 46 and 51: "... selector referencing data that does not exist in IP packets..."

Claims 47 and 52: "... determining a task ID..." and more.

Claims 48 and 53: "... determine a target application..."

Applicants, therefore, urge that the rejections under 35 U.S.C. 102 be withdrawn, and claims 1-14, 16-27, 29-32, 34, and 36-53 be allowed.

35 U.S.C. 103

Claims 15, 28, 33 and 35 have been rejected over Fiveash in view of Cunningham, et al. (U.S. Patent 6,219,786, hereinafter Cunningham).

Applicants traverse, and argue that the Examiner has not established a prima facie case of obviousness.

Each of these claims is a dependent claim, and is distinguished from Fiveash as discussed previously with respect to their respective base claims and, as will be discussed hereafter, these claims recited limitations not suggested by the combination of Fiveash and Cunningham.

In numbered paragraph 12 of the Office Action, the Examiner makes no specific references to Fiveash, Cunningham nor the present applciation. However, in response, applicants assert that Cunningham requires putting a workstation on a network (e.g. LAN) in promiscuous mode so that it can obtain all the IP packet traffic on the LAN (col 3, 35-55 & Fig 1). This also see by claim 1 clause 1 (col 11, 60-63); "...monitoring network traffic, including receiving data packets transmitted to and from nodes of said network such that receptions of said data packets are non-intrusive with respect to traffic flow of said network;".

Hence, a very basic and important distinction between the present application and Cunningham is that Cunningham clearly uses only IP packet data (as it must since the only packets that can appear on the networks in question are IP packets). Other differences are that Cunningham rules processing does not occur in the operating system kernel of the sending and receiving network nodes, and for inbound processing Cunningham does not have any kind of look-ahead function. The present application does not involve any kind of "assembling the data packets" (Cunningham col 8, lines 9-24).

Regarding claim 15, Cunningham does mention "time-slot specifies" (col 8, 61-62, a citation not made by the Examiner). While the combination of Fiveash & Cunningham may be apparent with respect to time-of-day indicia, this claim depends from claim 10 and is distinguished from the combination based on the claim 10 recitation 'non-IP packet data accessed with said system kernel', as previously discussed.

Regarding claims 28, 33 and 35, as noted above, Cunningham does not operate in the kernel of the operating system of the network node receiving the packet (even if it does reside in the kernel of what Cunningham calls the 'access management module' (Fig 2, item 26)). Said another way; Cunningham's rules processing is external ('non intrusive') to the network nodes being monitored (fig 2, col 6 49-67). Hence Cunningham cannot pass "said packet to said sockets layer", since that socket layer and the filtering in the present application, reside in the kernel of the operating system that received the packets. Neither can Cunningham mark "said packet as non-deliverable", again because such function resides in the communication protocol stack within a given node and involves passing data from a lower level (the filtering at the IP implementation) to an upper level (sockets). Since such marking & passing are technically impossible in Cunningham, and because Cunningham only deals with IP packets and data contained in those packets, it would not be obvious to anyone looking at Cunningham, to do such marking & passing. Hence claims 28, 33 & 35 are not obvious with respect to the combination of Cunningham & Fiveash.

Applicants, therefore, urge that the rejections under 35 U.S.C. 103 be withdrawn, and claims 15, 28, 33 and 35 allowed.

SUMMARY AND CONCLUSION

Applicants urge that the above amendments be entered and the case passed to issue with claims 1-53.

The Application is believed to be in condition for allowance and such action by the Examiner is urged. Should differences remain, however, which do not place one/more of the remaining claims in condition for allowance, the

Examiner is requested to phone the undersigned at the number provided below for the purpose of providing constructive assistance and suggestions in accordance with M.P.E.P. Sections 707.02(j) and 707.03 in order that allowable claims can be presented, thereby placing the Application in condition for allowance without further proceedings being necessary.

Sincerely,

Edward B. Boden

By

Shelley M Beckstrand

Shelley M Beckstrand
Reg. No. 24,886

Date: 3 February 2005

Shelley M Beckstrand, P.C.
Attorney at Law
61 Glenmont Road
Woodlawn, VA 24381-1341

Phone: (276) 238-1972
Fax: (276) 238-1545